AD-A031 210 FEDERAL AVIATION ADMINISTRATION WASHINGTON D C SYSTE--ETC F/G 17/7
CENTRAL FLOW CONTROL COMPUTER PROGRAM SPECIFICATIONS. VOLUME IV--ETC(U)
SEP 76 R BALES, G BEEKER, C BROGLIO, F CASEY

UNCLASSIFIED FAA-RD-76-157-VOL-4 NL

1 OF 1
AD A031210

END

DATE FILMED
11-76

Report No: FAA-RD-76-157, IV

DDC 12

CENTRAL FLOW CONTROL COMPUTER PROGRAM SPECIFICATIONS: VOLUME IV DATA BASE SUBSYSTEM SPECIFICATION

Central Flow Control Design Team
Federal Aviation Administration

September 1976
Final Report

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
FEDERAL AVIATION ADMINISTRATION
Systems Research & Development Service
Washington, D.C. 20590

NOTICE

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FAA-RD-76-157 IV-Vol-4		
4. Title and Subtitle		**5. Report Date**
Central Flow Control Computer Program Specifications, Volume IV. Data Base Subsystem Specification.		Sept 1976
		6. Performing Organization Code ARD-102
7. Author(s)		**8. Performing Organization Report No.**
Central Flow Control Design Team		111-102
9. Performing Organization Name and Address		**10. Work Unit No. (TRAIS)**
U.S. Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington, D.C. 20590		111-102 **11. Contract or Grant No.** 111-102
12. Sponsoring Agency Name and Address		**13. Type of Report and Period Covered**
U.S. Department of Transportation Federal Aviation Administration Systems Research and Development Service Washington, D.C. 20590		Final Report,
		14. Sponsoring Agency Code ARD-102

15. Supplementary Notes

R. Bales, G. Beeker, C. Broglio, F. Casey Y. Chao

73 p.

16. Abstract

This report contains the specifications for the data base subsystem of the Central Flow Control Computer Program. This report provides a detailed functional specification of the system's database management components, and a detailed specification of the database structure.

17. Key Words	18. Distribution Statement
File Management, Database Management, Functional Specifications, Logical File Handlers, Flow Control, Offical Airline Guide	Document is available to the public through the National Technical Information Service, Springfield, Virginia 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
UNCLASSIFIED	UNCLASSIFIED	78	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

340 170 LB

METRIC CONVERSION FACTORS

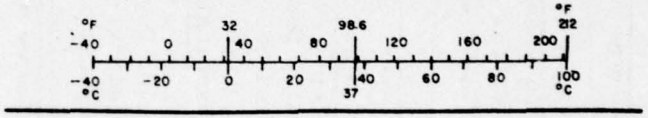
Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
	AREA			
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
	MASS (weight)			
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
	VOLUME			
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
	TEMPERATURE (exact)			
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.

Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
	LENGTH			
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
	AREA			
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
	MASS (weight)			
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
	VOLUME			
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
	TEMPERATURE (exact)			
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



LIST OF MEMBERS OF THE CENTRAL FLOW CONTROL DESIGN TEAM

	R. Bales	MITRE	T. McCann	AAT-370
	G. Beeker	MITRE	M. McCrory	ARD-141
(1)	C. Broglio, Ph.D.	ARD-102	M. Medeiros	TSC-622
	F. Casey	ARD-141	O. Morganstern	MITRE
	Y. Chao, Ph.D.	FEDSIM	J. Noyes	AAF-630
	R. Davis	FEDSIM	W. Reed	AAF-643
	G. Ellison	ARD-142	J. Richardson	AAT-370
	H. Gabrielli	MITRE	A. Robb	TSC-622
	T. Guyc	BDM	D. Rozzano	ARD-142
	V. Hallows	AAT-510	A. Severino	ARD-142
(2)	T. Hannan	ARD-102	J. Siedsma	AAT-510
	L. Jarze	FEDSIM	R. Watson	MITRE
	G. Kershaw	BDM	H. Whang	TSC-622
	P. MacDonald	TSC-622	G. Wright	FEDSIM
	V. Maglione	TSC-622	R. Wright	TSC-622

(1) Chief Programmer
(2) Back-up Programmer

ACKNOWLEDGEMENT

This specification series was prepared by the members whose names are listed above. In performing this work these members were supported by their respective organizations. Inclusion in the above list, however, does not necessarily imply that either the individual or his organization endorses the report.

CONTRIBUTING ORGANIZATIONS

AAF - Airway Facilities Service, Federal Aviation Administration
AAT - Air Traffic Service, Federal Aviation Administration
ARD - Systems Research and Development Service, Federal Aviation Administration
BDM - The BDM Corporation, Vienna, Virginia
FEDSIM - Federal Computer Performance Evaluation and Simulation Center
MITRE - The MITRE Corporation, McLean, Virginia
TSC - Transportation Systems Center, Department of Transportation

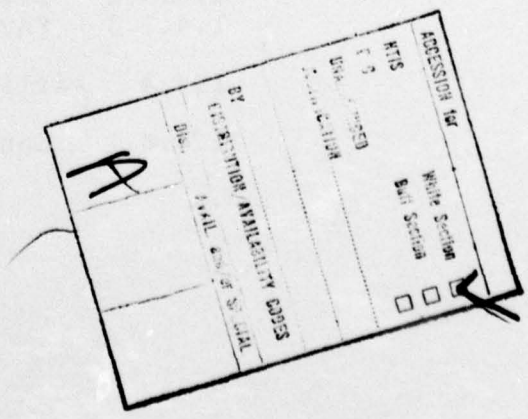


TABLE OF CONTENTS

TABLE OF CONTENTS (Cont.)

TABLE OF CONTENTS (Cont.)

LIST OF ILLUSTRATIONS

INTRODUCTION

The data on scheduled and non-scheduled flight information between departure and arrival terminals is the heart of the Central Flow Control system. The flight information is initially obtained from the Official Airline Guide (OAG) tapes. Subsequently, this data is updated and complemented to provide an up-to-date status of each of the flight legs.

While such flight leg information is the primary component of the Central Flow Control (CFC) data base, other files are necessary to provide data on the Air Traffic Control environment (e.g., airports, centers, zones, etc.), and to support search and access operations (e.g., indexing). The data base is defined as the collection of all files whether disk-resident or in main memory, containing flight leg information and the supporting files as described above. These files are described in detail in Section 1.

Specification of the Data Base Subsystem involves two major areas. First, the structure of the data base is defined (section 1) describing the physical and logical characteristics of the individual files contained in the data base. Second, the data base management functions are specified. This involves the processing, the program logic and the interfaces required in order to access the data base, for reading and updating the data base contents, and to accomplish the necessary maintenance functions. The data base management requirements are specified in Section 2.

The interrelationships between the on-line and the off-line support system in reference to the data base are discussed in Appendix A. Appendix B provides a summary describing how the application programs make use of the data base management facilities for data base access.

1. DATA BASE STRUCTURE

The Central Flow Control data base consists of the following files and tables:

1. (1) OAG Flight Record File (OFRF)
2. (1) Non-Scheduled Flight Record File (NSFRF)
3. Flight Index File (FIF)
4. General Aviation Table (GAT)
5. Capacity Table (CAT)
6. Arrival/Departure Table (ADT)
7. Airline Table (AOT)
8. Flight Accession Table (FAT)
9. Zone Table (ZOT)
10. Continue Table (COT)
11. Arrival Fix Table (FXT)
12. Airport/Fix Table (AFT)
13. Aircraft Type Table (ATT)
14. Aircraft Class Table (ACT)
15. Center Table (CET)
16. Airport Table (APT)
17. Table Mapping Table (TMT)
18. Conversion Dictionary Table (CDT)
19. System Statistics Table (SST)
20. Operational Category Table (OCT)
21. Output Format Table (OFT)
22. Output Device Table (ODT)
23. Parameter Table (PAT)
24. Non-OAG Name Table (NOT)

For each of the above components of the data base the following specifications include: definition of the data fields, estimated sizing information and conceptual description of the physical record management.

In general, within a particular file, all physical records are fixed length. Logical records are usually fixed length, but in some files they may be variable length.

(1) The above first two flight record files, are logically treated as two distinct files, but may be implemented as one or two physical files.

1.1 OAG Flight Record File (OFRF)

The OFRF is constructed from R. H. Donnelley's Official Airline Guide (OAG). The initial building of the OFRF is performed by the off-line support system. In the course of normal CFC operations the OFRF will be changed and updated on-line based on data received from the NAS ARTCCs and the SCC. OFRF contains data for 30 (SP)[1] days of operations. Every record within OFRF contains information on a single flight leg between two terminals (i.e., an arrival and a departure terminal).

1.1.1 Data Fields

The basic unit of granularity in each flight record will be one byte. The data will be stored in internal code form to facilitate searching. All alphanumeric data (e.g., ACID, airport id) will be converted from characters to the internal representation using appropriate dictionaries for this purpose. All time data will be expressed in seconds.

Each logical flight record includes the following fields:

1. Relative Address

   The address of the record relative to the beginning of the file.

2. Aircraft Identification (ACID)

   The ACID field consists of two subfields:

   a. operator code
   b. accession ID

   ACID will be stored in internal code form to facilitate searching.

3. Departure Terminal

   The pacing airport or the center where this flight leg originates.

(1) (SP) is an abbreviation for System Parameter.

4. Arrival Terminal

The pacing airport or the center where this flight leg terminates.

5. Scheduled Departure Time (SDT)

6. Estimated Time Enroute (ETE)

7. Aircraft Type, Class and Category

For example, this field may include the internal representation for: B747, Jet, Aircarrier

8. Departure Date Mask

A mask specifying, for each day of the current month, if this flight is scheduled to depart or if it is not scheduled.

9. Activation/Deactivation Date Mask

A mask specifying, for each day of the current month, if this departure is activated or deactivated.

10. Arrival Date Mask

Same as 8 above, but pertains to the arrival date.

11. Actual Departure Time

Departure time as reported by a NAS DM (Departure Message).

12. Controlled Departure Time

Departure time as calculated based on flow control information

13. "Best Estimate" Key

This key denotes which departure time (scheduled actual or controlled) was used to arrive at the "Best Estimate" of Arrival Time (see 14 below).

14. "Best Estimate" of Arrival Time

This field contains scheduled arrival time initially, and can be subsequently modified by actual departure time plus ETE or by a controlled departure time plus ETE.

15. Special Simulation Data

This data field is provided for the exclusive use of the simulation subsystem.

16. Status

This field includes the following status information:

- null or deleted record
- record derived directly from OAG tape
- record reflects a change to OAG schedule
- new record not in OAG schedule
- pointer to next record in this group (OAG schedule or a change)

Other status information may be included in this field as needed.

1.1.2 Sizing

Each logical flight records will require approximately 20 words of storage. The number of logical flight records in OFRF is estimated at 24,000 initially, but should be expandable to make provisions for future needs.

1.1.3 Physical Record Arrangement

The OFRF in its entirety is stored on disk. All flight records between a unique pair of terminals constitute a "record group". The record groups are blocked with each record group being physically distinct. A record group may occupy multiple physical blocks; however, no flight record is split between two physical blocks (see Figure 1-1).

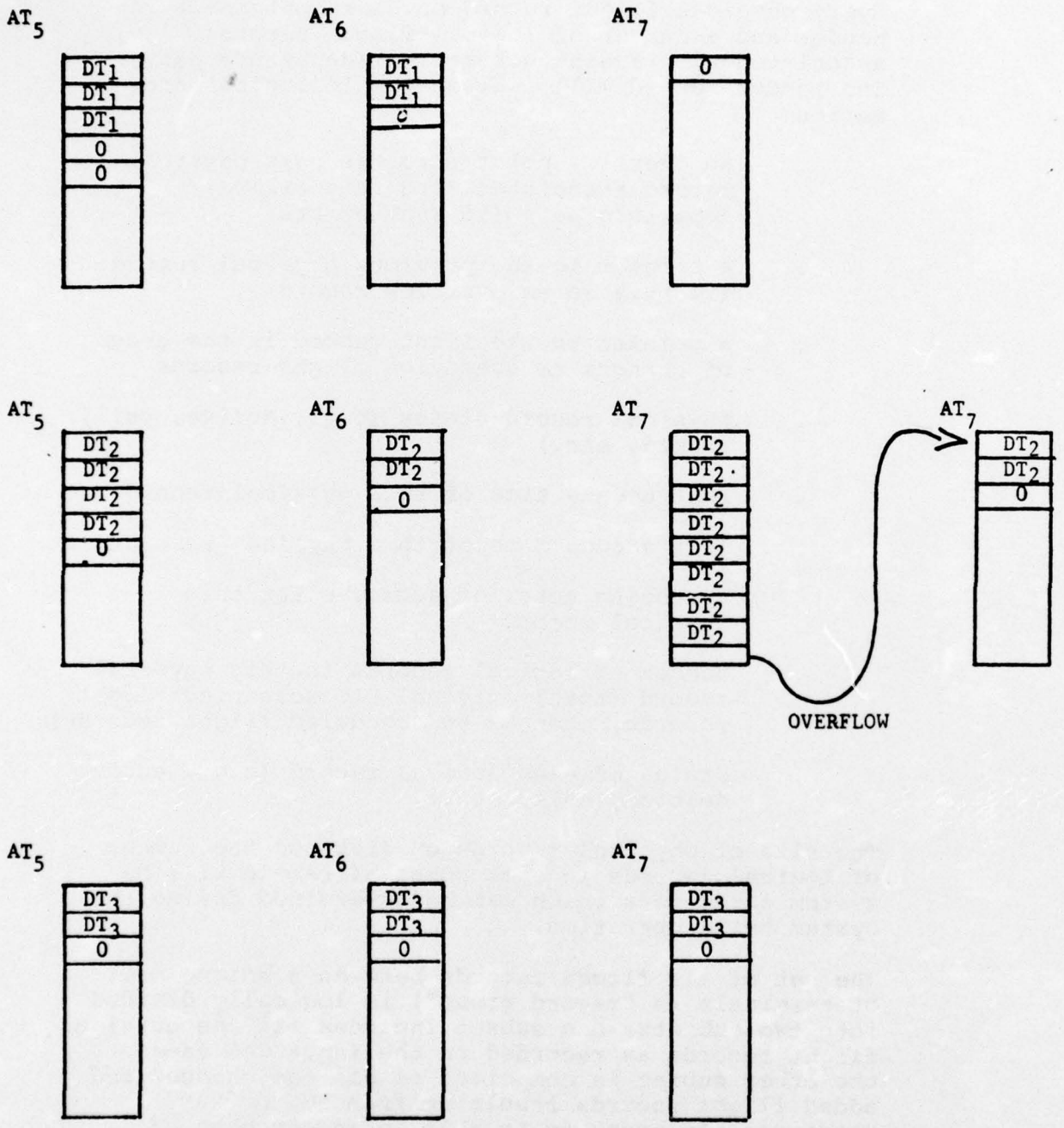


FIGURE 1-1

ILLUSTRATION OF FLIGHT RECORD FILE PHYSICAL ARRANGEMENT
(only a portion of the file is shown)

AT - arrival terminal
DT - departure terminal
0 - denotes end of record

Every physical flight record on disk contains a header and a number of logical flight records associated with a distinct arrival/departure pair. The header record will include the following information:

- an overflow pointer to the next physical record associated with this arrival/departure pair (if such exists)
- a pointer to the previous physical record (if this is an overflow record)
- a pointer to the first record in the group of changes to scheduled flight records
- physical record status (e.g., active, null, locked, etc.)
- last update time of this physical record
- last access time of this physical record
- a running total of accesses for this physical record
- number of logical records in this physical record (total original OAG scheduled flight records, changes to scheduled flight records)
- status of each logical record (e.g., active, deleted, null, etc.)

The size of physical records on disk and the number of logical records in each physical record will be system parameters which can be determined during the system build operation.

The set of all flight records between a unique pair of terminals (a "record group") is logically divided into two subsets; one subset includes all the original flight records as recorded on the input OAG tape; the other subset is comprised of all the changed and added flight records resulting from SCC or NAS messages. In order to be able to access each of these subsets individually, the header record contains two pointers corresponding to the first

flight record in each of the above subsets. In addition, each record contains a pointer to the next sequential record in its subset. Record counts and last record flags are also provided.

1.2 Non-Scheduled Flight Record File (NSFRF)

The NSFRF is a distinct file containing non-scheduled flight records. These flights normally apply to non-aircarrier operations. This information is stored in NSFRF during on-line operations based on data received primarily from the NAS ARTCCs and occassionally from the SCC. The off-line support system initially builds the NSFRF but no data is placed by the off-line system in the records.

1.2.1 Data Fields

Same as specified for OFRF (Section 1.1).

1.2.2 Sizing

The size of each logical record is approximately 20 words. The total number of logical records is initially estimated as 12,000 but should be easily expandable.

1.2.3 Physical Record Arrangement

Conceptually the same as specified for OFRF (Section 1.1), but the grouping of logical records within a physical record might be different to achieve better disk utilization.

1.3 Flight Index File (FIF)

The FIF is a disk-resident file which is used by DBMS in order to locate flight records based on a given ACID. FIF is normally accessible only to the Logical File Handlers (LFHs). This will insure that possible changes to the FIF structure do not require major changes to the application program.

1.3.1 Data Fields

Each logical record in FIF contains the following fields:

1. Aircraft Identification (ACID)

   ACID is stored in internal code form.

2. Relative Location

   The address of the flight record relative to the beginning of the file.

3. File Identification

   Points to the file in which this flight record can be found.

4. Next ACID

   Points to the location of the next ACID entry in the FIF.

5. Next Flight Leg

   Points to the location of the next flight leg entry in the FIF for this ACID.

6. Arrival Terminal

   Identifies the pacing airport or the center.

7. Departure Terminal

   Identifies the pacing airport or the center.

1.3.2 Sizing

Each logical record in FIF will require approximately four words of storage.

1.3.3 Physical Record Arrangement

The FIF is variable length; the record groups are blocked. The record groups contain the flight number indices for an airline or an aircraft identifier for other aviation. A record group may occupy multiple physical record; however, no flight number index record may be split between two physical records.

The FIF is a linked list of index records which contain the relative file locations of flight records in the flight record file. Each record contains a relative file location of the flight record of an individual flight leg.

A table containing the location of a block of flight numbers is associated with airline or aircraft identifiers. The block of flight numbers contains the head of a chained list of flight number indicators. The flight number indicators point to the beginning of a sequence of flight number index records corresponding to all flight records associated with a flight number.

The relationship of the links and pointers is presented in Figure 1-2. The construction of the sequencing of the flight number index records is shown in Figure 1-3. Airline indicators point to the flight number indicators directly, if the number of flights for that airline is small. This is indicated in Figure 1-2 by a null pointer for airline indicator N-1.

1.4 General Tables

The general tables which are part of the CFC data base structure are specified in this section. Some of the tables are accessible by the application programs through the GET TABLE and the SET TABLE routines (Section 2). Other tables are used to support the functions of DBMS.

1.4.1 General Aviation Table (GAT)

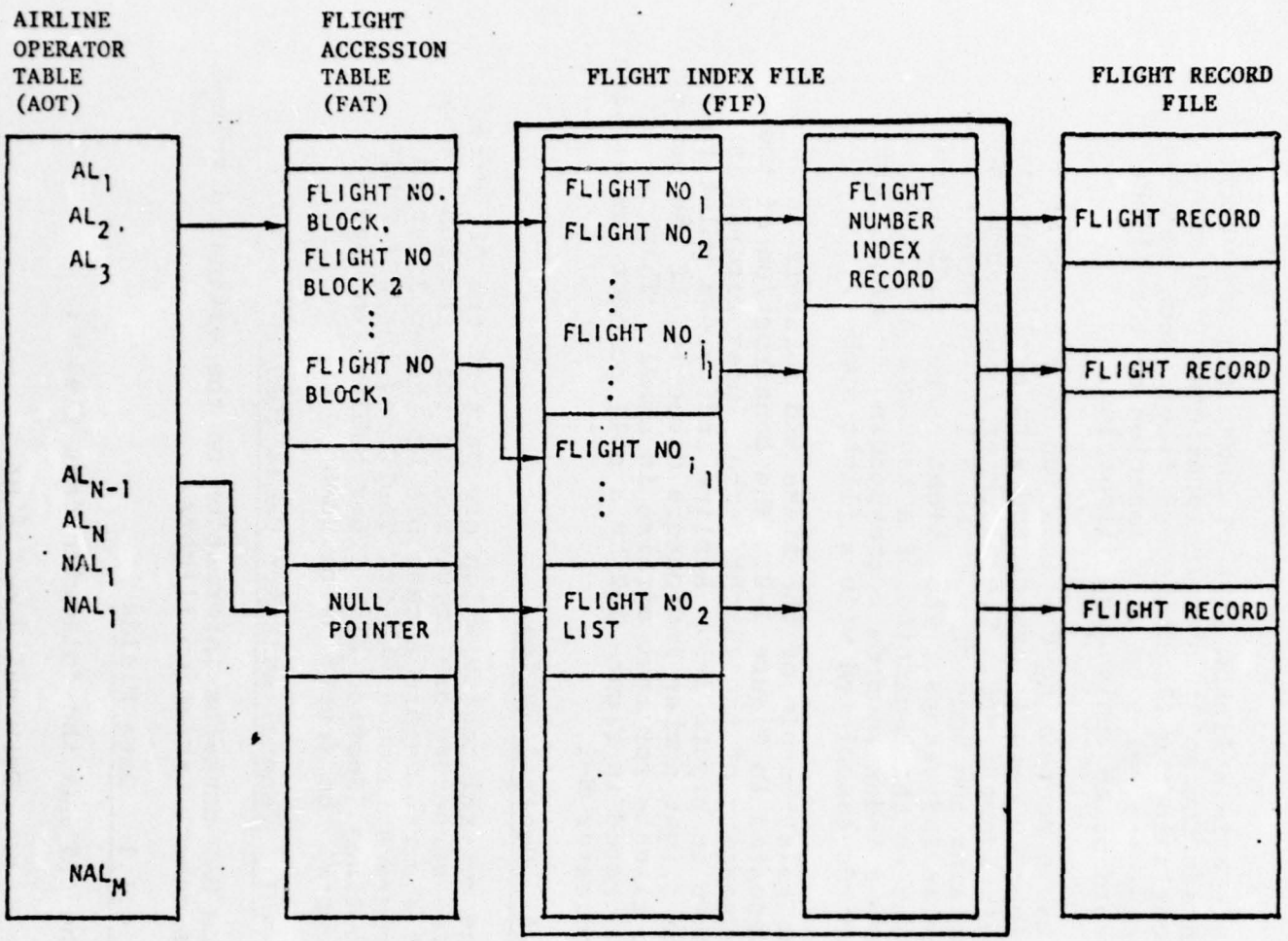
The GAT contains information on the estimated volume of general aviation flights.

1.4.1.1 Data Fields

The GAT has the following data fields:

1. Terminal Identifier

   Pacing terminal or center



NON-SCHEDULED AVIATION = NAL

FIGURE 1-2

FLIGHT NUMBER IDENTIFIER INVERTED LIST AND LINK LIST STRUCTURE

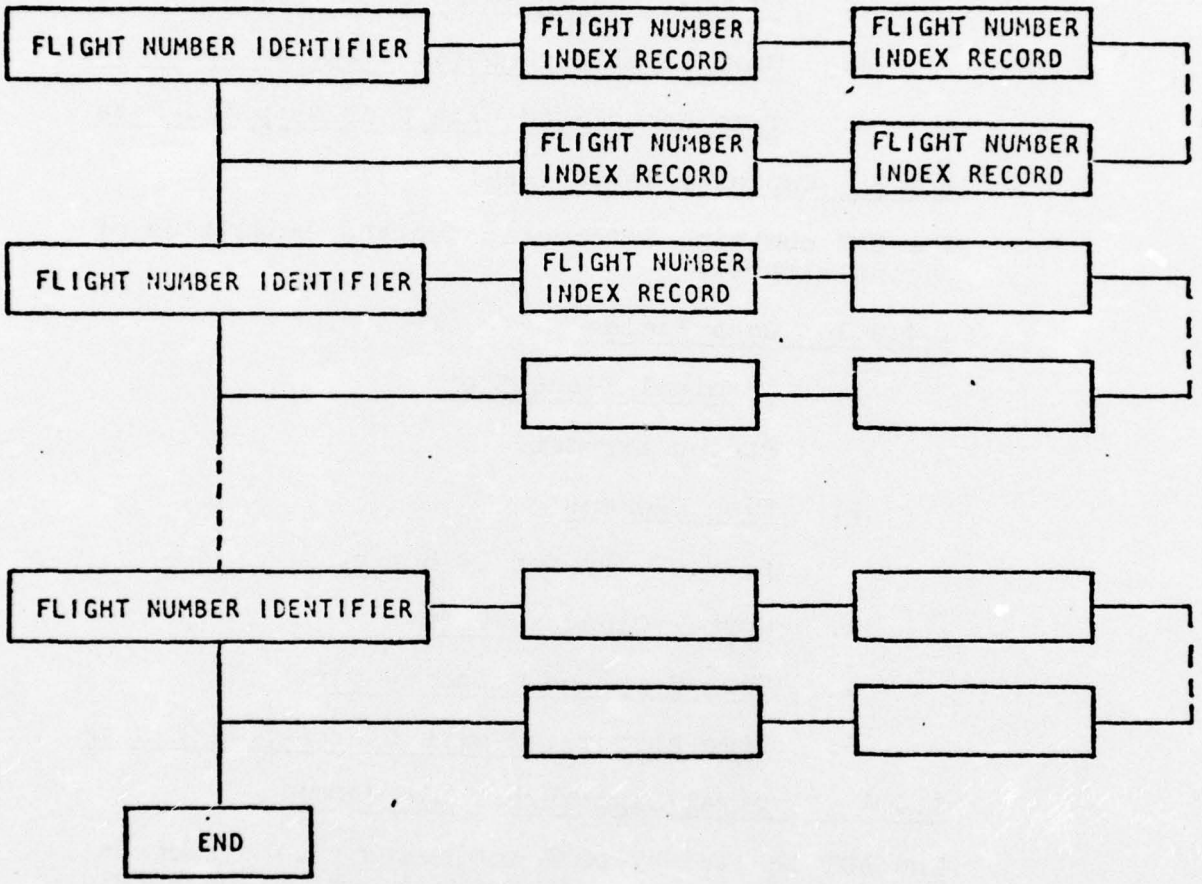


FIGURE 1-3

FLIGHT INDEX FILE BLOCK STRUCTURE

2. Time Segment

Hour of day for 24 hours

3. Normal General Aviation Estimate

4. User Supplied General Aviation Estimate

5. Time Associated with User Supplied Data

1.4.2 Capacity Table (CAT)

The CAT contains information on the capacities of pacing airports.

1.4.2.1 Data Fields

1. Terminal Identifier

Pacing airport

2. Time Segment

Hour of day for 24 hours

3. Normal Capacity Term

4. User Supplied Capacity Term

5. Time Associated with User Supplied Data

1.4.3 Arrival/Departure Table (ADT)

The ADT is used by DBMS to locate flight records based on arrival or departure location (or both). This table relates the location of blocks of flight records to the arrival and departure pacing airports or centers.

The ADT is a two dimensional array. Columns correspond to arrival terminals and centers; rows correspond to departure terminals and centers. The array is depicted in Figure 1-4 in which pacing airports are represented by A and centers are represented by T. The intersections of columns and rows contain the relative file locations of the flight records between arrival and departure terminals and centers. The elements in the matrix contain the relative locations of the flight record

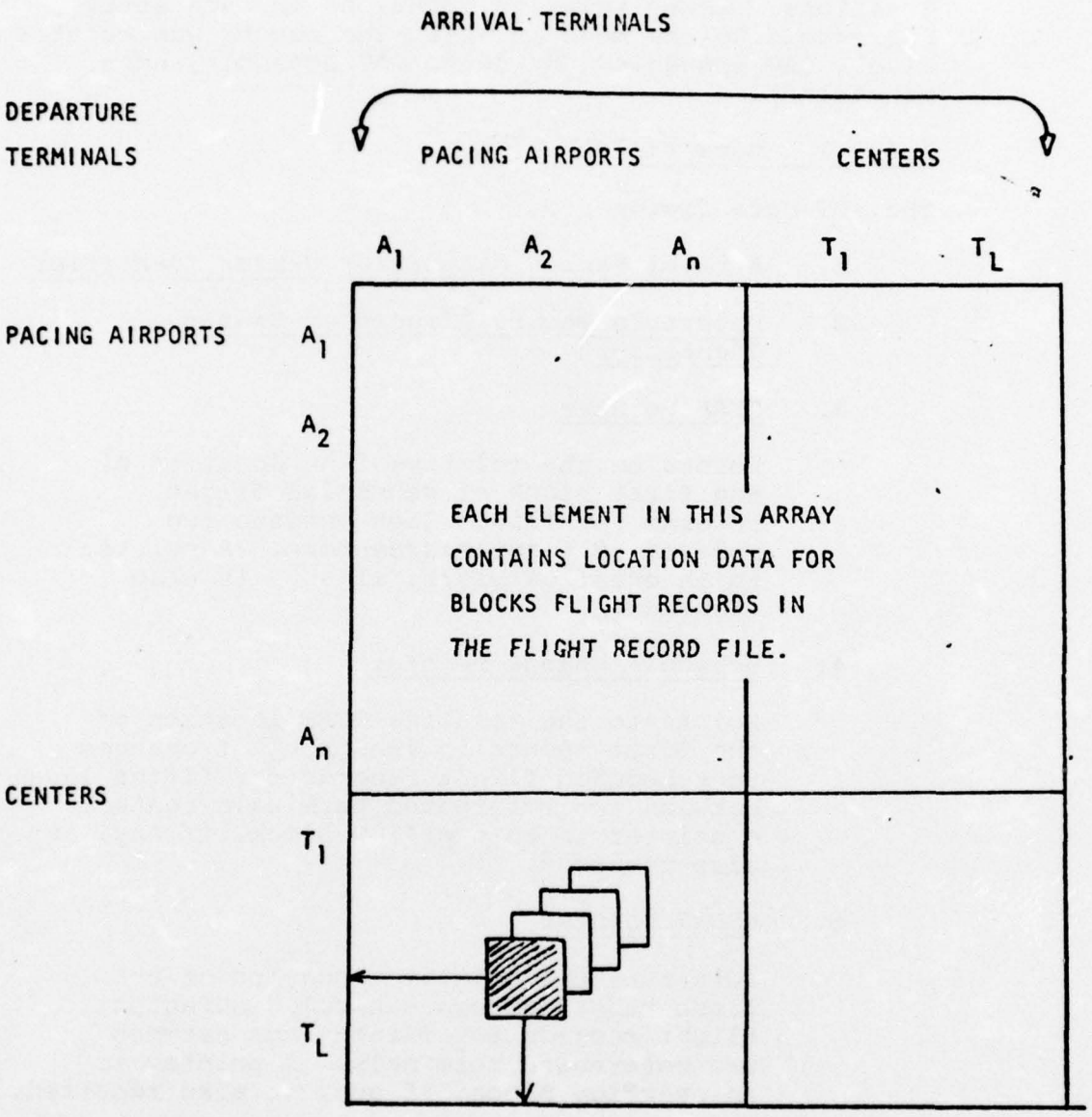


FIGURE 1-4

ARRIVAL/DEPARTURE TABLE (ADT) ORGANIZATION

blocks of air traffic between the arrival and departure locations. Three relative locations in each entry correspond to the mode in which the record was created (i.e., OAG schedule, change to OAG schedule, non-scheduled).

1.4.3.1 Data Fields

The ADT data fields are:

1. Arrival Pacing Airport or Center Identifier

2. Departure Pacing Airport or Center Identifier

3. OFRF Pointer

   Points to the relative file location of the first block of scheduled flight records for flight legs between two referenced terminals/centers. A pointer to an overflow block, if any, is also required.

4. Schedule Change Pointer

   Points to the relative OFRF location of the first record in the group of changes to scheduled flight records for flight legs between two referenced terminals/centers. A pointer to an overflow block, if any, is also required.

5. NSFRF Pointer

   Points to the relative location of the first record of non-scheduled aviation flight records for flight legs between two referenced terminals. A pointer to an overflow block, if any, is also required.

6. Block Count

   Contains the count of blocks in each of the data fields 3, 4 and 5 above.

1.4.3.2 Sizing

Each cell in the ADT will contain the above data fields. The number of cells will be: 1600 corresponding to 40 arriving and 40 departing locations. The 40 locations include the 15 pacing airports and 25 domestic centers and foreign areas. Each of these cells constitutes a fixed size logical record.

1.4.3.3 Physical Record Arrangement

The ADT utilizes a linked list organization as represented in Figure 1-5.

The elements in the two dimensional array structure of the ADT can be searched according to the following keys:

- Pacing arrival airport or an arrival center (for non-pacing airport)
- Pacing departure airport or a departure center (for non-pacing airport)

1.4.4 Airline Operator Table (AOT)

The AOT contains a list of valid aircarrier/airtaxi codes, and their corresponding internal representation. In addition, AOT contains pointers to the Flight Accession Table (FAT, Section 1.4.5).

1.4.4.1 Data Fields

1. Operator Code

   The alphanumeric code of a valid aircarrier/airtaxi operator.

2. Internal Pepresentation

   The internal representation of the Operator Code.

3. Flight Accession Table (FAT) Pointer

   Points to the relative location of flight accession codes in FAT for this operator.

ARRIVAL TERMINAL$_1$

POINTER TO NEXT ARRIVAL TERMINAL SEGMENT

DEPARTURE TERMINAL$_1$

POINTER TO NEXT DEPARTURE TERMINAL GROUP

FLIGHT RECORD,

.

.

.

FLIGHT RECORD N_1

DEPARTURE TERMINAL$_2$

POINTER TO NEXT DEPARTURE TERMINAL GROUP

FLIGHT RECORD,

.

.

.

FLIGHT RECORD N_2

ARRIVAL TERMINAL$_2$

.

.

.

ARRIVAL TERMINAL$_N$

FIGURE 1-5

LINKED LIST ORGANIZATION USED IN THE
ARRIVAL/DEPARTURE TABLE (ADT)

1.4.5 Flight Accession Table (FAT)

The FAT contains unique or groups of flight accession codes and pointers to the flight record files and the Flight Index File (FIF). The entries in FAT include all flight numbers for scheduled flights or identifiers for non-scheduled flights.

1.4.5.1 Data Fields

1. Accession Code Range

   This field identifies a range of accession codes for one operator (e.g., 100-199 for TWA). A unique accession code can also be specified.

2. OFRF Pointer

   Pointer to the first relative location in OFRF corresponding to the entry in field 1.

3. NSFRF Pointer

   Points to the first relative location in NSFRF corresponding to the entry in field 1.

4. FIF Pointer

   Points to the first relative location in FIF corresponding to the entry in field 1.

1.4.6 Zone Table (ZOT)

The ZOT is used only by the Simulation Subsystem. This table specifies the relationship between terminals in a zone designation.

1.4.6.1 Data Fields

1. Pacing Airport

2. Pointer to Next Pacing Airport

3. Pointer to Next Zone for this Pacing Airport

   Up to five zones may be specified for a pacing airport.

4. Zone Data

This group of data includes the following zone information:

a. nominal flight time for general aviation
b. a tier center
c. flight time between the tier center boundary crossing and the destination airport
d. origin center identifiers for this tier center

1.4.7 Continue Table (COT)

The COT is used only by the Simulation Subsystem. This subsystem also supplies and updates the data contained within COT.

The COT is ordered by pacing airport identifier. Other sequence key are established by the simulation program.

1.4.8 Arrival Fix Table (FXT)

The FXT specifies the relationship between the fixes and the airports. A maximum of 256 fixes are accommodated.

1.4.8.1 Data Fields

1. Arrival Fix Identifier

2. Number of Pairs of Arrival and Departure Pacing Airports

3. Arrival Pacing Airport

   The Pacing Airport associated with the arrival fix.

4. Departure Pacing Airport

   Lists all departure airports whose flights will arrive at the arrival airport via the listed arrival time.

5. Value

   This item is a percentage of FTE expired at this fix.

1.4.9 Airport/Fix Table (AFT)

For each pacing airport, the list of arrival fix codes is specified.

1.4.91 Data Fields

1. Pacing Airport
2. Number of Fixes
3. Fix Identifier

1.4.10 Aircraft Type Table (ATT)

The ATT contains the designators for valid aircraft types (e.g., B747, DC10, etc.) and associates each of these types with the aircraft class (i.e., Jet, Prop, Turbo, not specified). ATT also provides the internal representations of the aircraft type.

1.4.11 Aircraft Class Table (ACT)

The ACT contains the designations for valid aircraft class (i.e., Jet, Prop, Turbo, not specified) and for each class lists the associated aircraft types (e.g., B747, DC10, etc.). ACT also provides the internal representations of the aircraft class.

1.4.12 Center Table (CET)

This table contains the designations for all valid centers and the associated pacing airport codes and pacing airport indicators followed by all other airport codes for the center. A maximum of 20 centers will be accommodated.

1.4.13 Airport Table (APT)

This table contains the designators for all valid airports, whether pacing or non-pacing, and the associated valid center. Also, the internal representation of each airport and center is provided together with a Pacing airport indicator. Pacing airports are listed first. A maximum of 1200 airports are accommodated.

1.4.14 Table Mapping Table (TMT)

The TMT provides the Data Base Subsystem with control information related to the mapping of various components of the Data Base, such as the memory addresses of the tables, their disk addresses (if disk-resident) and their sizes.

1.4.15 Conversion Dictionary Table (CDT)

The CDT contains a set of dictionaries required to support the Data Base Subsystem operations. These tables are used to convert external representation of airports, centers, zones, fixes, flight identifiers, operator codes, and aircraft characteristics to internal codes and conversely.

1.4.16 System Statistics Table (SST)

System statistics on the use of the data are maintained in the SST. The statistics provide logical and physical record utilization for all files in the Data Base. The SST is continually updated by the Data Base Subsystem, to provide up to date performance measures.

The elements of the SST include the following data:

- File identifier
- Record identifier
- Number of times referenced
- Cumulative, maximum and running average of wait time
- Cumulative, maximum and running average of seek time

1.4.17 Operational Category Table (OCT)

The operational category table contains the legal operational category codes i.e., (air carrier (C), airtaxi (T), military (M), general aviation (G)).

1.4.18 Output Format Table (OFT)

The output journal table contains the valid output format codes (i.e., DEP, PGTD, ARR, PTOA, CTR, Type or Class, ETE).

1.4.19 Output Device Table (ODT)

The output device table contains the legal output device identifier and the appropriate routing code for each output device.

1.4.20 Parameter Table (PAT)

The parameter table contains values for the stop time parameter, stop data parameter, delay factor limit, ETE limit, stack time limits, hold time limit, capacity limit.

1.4.21 Non-OAG Name Table (NOT)

The non-OAG name table contains the legal alphabetic designator for all non-aircarrier call signs.

2. DATA BASE MANAGEMENT

The Data Base Management Subsystem (DBMS) is the collection of on-line programs required for retrieving, updating and maintaining the CFC data base. The components of DBMS include:

1. Logical File Handlers (LFHs)
2. Executive Interface Facilities
3. Control and Utility programs

2.1.1 Concept

The DBMS is designed to provide the following major capabilities:

1. facilitate application programs access to the data base

   DBMS provides the applications programs with conveniently useable file handlers designed to relieve the application programs from the chores of searching, accessing, checking and general housekeeping of the data base.

2. insure data base integrity

   DBMS is responsible for the data base and its components at all times. No change in the data base can take place other than through the facilities provided by DBMS. Every request for data base access and/or update is scrutinized by the DBMS and rejected if found illegal.

3. support test and evaluation functions

   DBMS provides the hooks for on-line recording of significant events and collecting of system performance measures.

2.1.2 Applications Program Interface

The LFHs provide the interface between the application programs and the data base. In fact, the application programs should never read and must not modify the data base other than through the services of the LFHs. The LFHs also perform error and authorization checking to insure the integrity of the data base.

The LFHs that are provided for the use of the application programs are described in Section 2.2.1. They are:

1. Get Record (GETR) - used to retrieve a flight record

2. Get Next Record (GETE) - used to retrieve a subsequent flight record after GETR or a previous GETE was used

3. Get Block (GETB) - used to retrieve a block of flight records

4. Get Next Block (GETN) - used to retrieve a subsequent block of flight records after GETB or a previous GETN was used

5. Change Record (CHGR) - used to change data elements in a flight record

6. Change Block (CHGB) - used to change data elements in multiple flight records

7. Insert Record (INST) - used to create a new flight record

8. Remove Record (REMR) - used to indicate that a flight record is no longer active

9. Get Table (TABT) - used to retrieve data elements from the General Tables

10. Set Table (SETT) - used to change data elements in the General Tables

Figure 2-1 depicts the CFC messages by the LFHs each invokes.

During the system build process, the LFH modules are appended as part of the Program Elements (PEs) taylored to process the various transactions. Upon execution of a specific transaction the LFH code is processed as part of the PE (Figure 2-2). The LFH communicates with the monitor requesting the search and access operations to read or write the physical data. Normally, unless error conditions were encountered, the data will be read into or written from a work area in main memory subject to processing by the application programs and/or the LFHs.

2.1.3 System Interface

Normally, access of the data base is invoked as a result of an input message from the SCC or by a CFC message from a NAS facility. On occassion, however, the data base must also be accessed in response to an internal system action or due to an external stimulus (e.g., system operator intervention). The above system action can be initiated on a cyclical time basis (e.g., recovery recording) or as a result of a special event (e.g., startover).

The processing associated with the system interfacing to the data base is illustrated in Figure 2-3. An appropriate Program Element (PE) is constructed for each transaction corresponding to a unique system function. The necessary LFHs are appended as part of such PE. The LFH issues the SVCs to invoke Executive action for the actual input and output processing of the data.

2.2 Logical File Handlers (LFHs)

The LFHs are programs which are used by the applications programs as well as by other software modules of DBMS. By calling an LFH the calling program invokes the processing needed to locate the desired records and to resolve the relationship between the logical and the physical addresses of the records. If the data does not reside in main memory, a LFH invokes the supervisor calls (SVCs) needed to access the peripheral storage media.

LOGICAL FILE HANDLERS \ MESSAGES	INHB	ACTV	FPSD	FP	CXSD	RS	DM	LIFP	LISA	LISD	DEMD	DEMA	DESD	DESA	DLDY	FIXL	ARRD	FADP	FADF	FADT	QFLZ	QFLW	CAPS	CAPL	GAES	GAEL
GET RECORD	•	•	•	•	•	•	•	•																		
GET BLOCK									•	•	•	•	•	•	•	•	•	•	•	•	•	•				
CHANGE RECORD	•	•	•	•			•																			
CHANGE BLOCK																		•	•							
INSERT RECORD			•	•																						
REMOVE RECORD					•	•																				
GET TABLE																								•		•
SET TABLE																							•		•	

FIGURE 2-1

MESSAGE PROCESSING RELATIONSHIP TO LOGICAL FILE HANDLERS

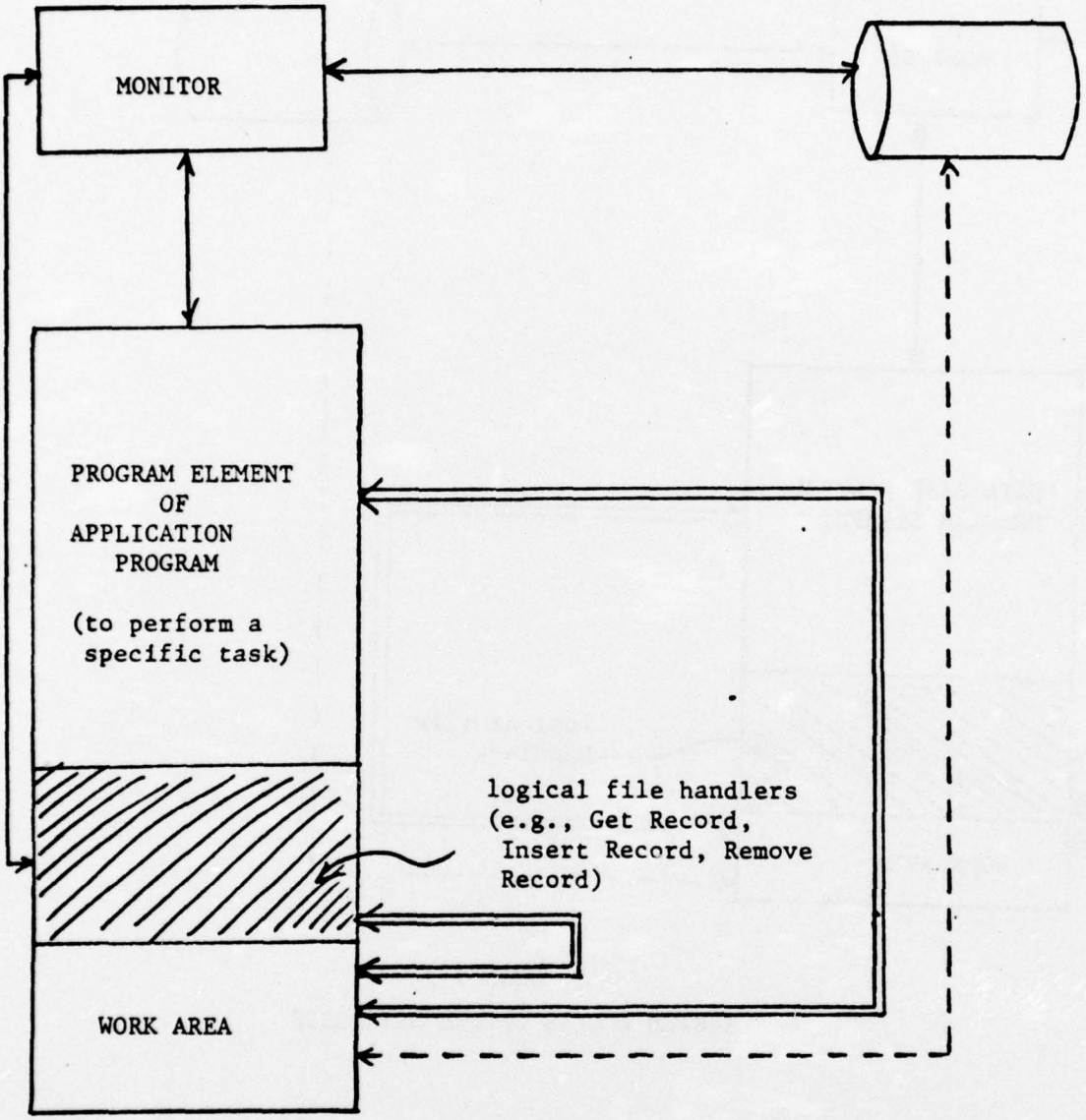
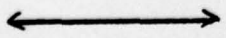


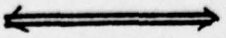
FIGURE 2-2

CONCEPT OF DB (DISK) ACCESS BY APPLICATION PROGRAMS



control path (e.g., SVC, commands)

data path



data manipulation

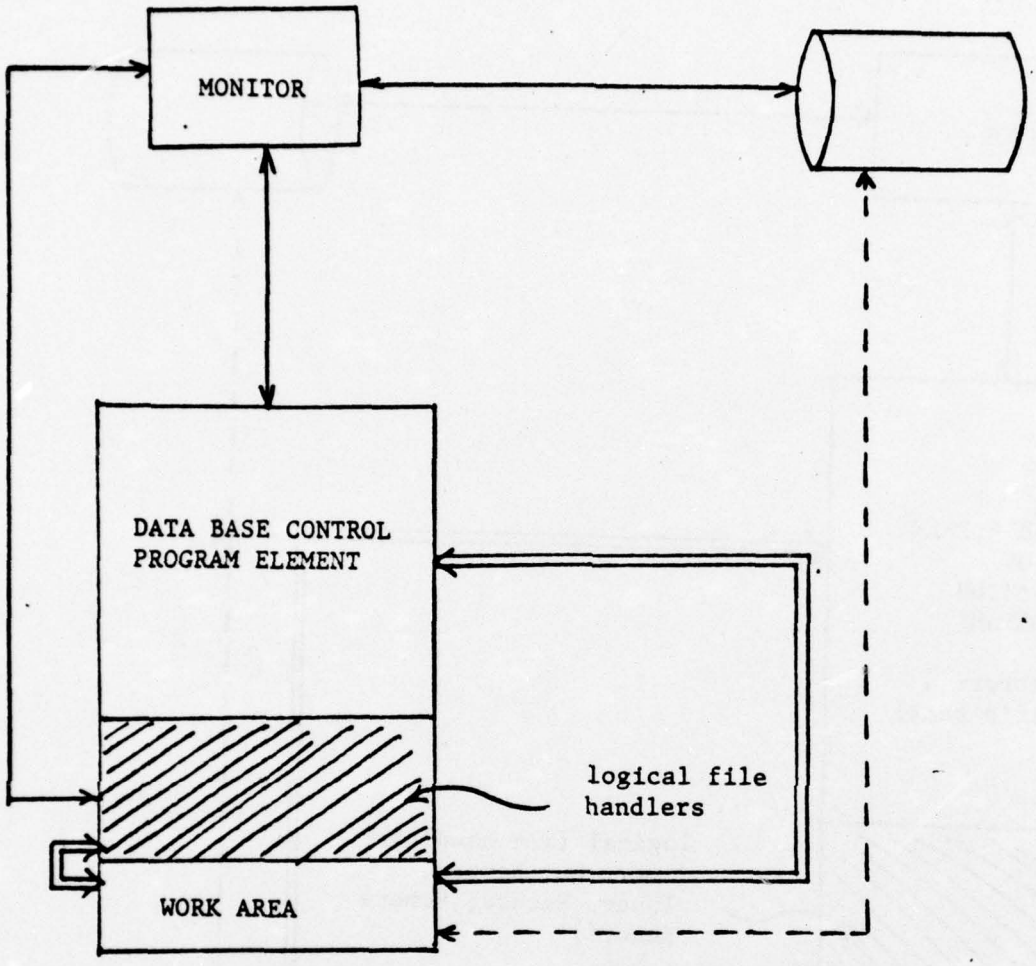
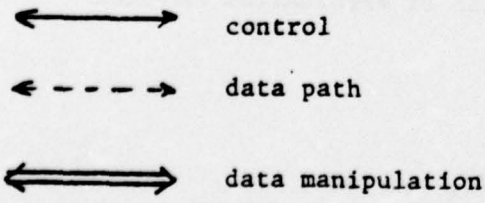


FIGURE 2-3

SYSTEM ACCESS OF THE DATA BASE



2.2.1 User Access to Flight Records

The following calls access the OFRF or the NSFRF: Get Record, Get Next Record, Get Block, Get Next Block, Change Record, Change Block, Insert Record and Remove Record.

2.2.1.1 Get Record (GETR, GETE)

a. Purpose

The Get Record routine is used to retrieve flight records using aircraft identification as the retrieval key. All flight records corresponding to the aircraft identification are returned to the user. Status indicators are set to describe the results of request activity. The GET Record routine has two entry points. GETR establishes the parameters for retrieving data and retrieving the first flight record. GETE is used to retrieve all subsequent flight records.

b. Users

Any message processing that uses aircraft identification as the primary key to identify the flight record to be retrieved must use this routine. In particular, this routine is required for message processing in which the final results contain a list of flight records by aircraft identification, i.e., LIFP.

Message processing and programs retrieving flight records by aircraft identification for the purpose of modifying data elements, including record insertions, must use this routine. The message processing programs using this routine are:

1. FPSD
2. CXSD
3. FP
4. DM
5. INHB
6. ACTV
7. RS
8. LIFP

c. Format of the Calling Sequence

Call GETR WA,

where WA is location of the start of user request sequence and contains retrieval parameters, status indicators and the location of user data block.

If the next record is requested, following a previous GETR or GETE call, the calling sequence is:

Call GETE WA,

where WA is the location of the user provided segment. The data segment contains status indicators and the location of the user data block.

d. Data Segment Definition

1. Aircraft Identification (ACID)
2. Arrival/Departure Location
3. Flight Record Qualifier
4. Request Status
5. Data Block Location

These are described below:

1. Aircraft Identification (ACID)

This information must be supplied by the user. Normally, both the operator code and the flight accession code will be provided. In special cases, a unique code identifying the flight will be provided.

If only the operator code is specified, with the flight accession code omitted, the first (or the next, if GETE) flight record for this operator will be retrieved.

2. Arrival/Departure Location

The following combinations of arrival/departure information can be specified by the user:

- arrival and departure location
- arrival location only
- departure location only
- no arrival location and no departure location

The arrival/departure location may be either a pacing airport or a center. The center designation references all the non-pacing airports within the specified center.

3. Flight Record Qualifier

The user may specify which of the following three types of flight records he wishes to access:

(i) scheduled flights
(ii) non-scheduled flights
(iii) scheduled and non-scheduled flights

If no such specification is provided, type (iii) above will be assumed.

4. Request Status

This data is provided by DBMS as a response to a GETR or GETE request. The following responses may be provided:

- error in data segment definition or illegal items in the data segment
- flight record not found
- flight record found and placed in data block location
- flight record file from which the flight record was retrieved (if record retrieval indeed took place)
- address of next record satisfying this request

5. Data Block Location

This information specifies the memory address of the flight record.

e. Interface

The tables used in the GETR/GNTE routine are:

- Airline Operator Table (ACT)
- Flight Accession Table (FAT)
- Flight Index File (FIF)

The routine uses disk open, close and read system routines. GIND (Section 2.2.3.1) utility routine is used. All requests are written to the data recording tape.

f. Processing

Figure 2-4 illustrates the sequence of tables referenced by DBMS in processing a GETR request where only the ACID is specified by the user.

Additional logic is required if the arrival/departure location is also specified by user in addition to the ACID. To accomplish this the LFH must check the arrival/departure data fields in the Flight Index File. If a match is found with the user requested arrival/departure location, the appropriate flight record will be retrieved from disk.

Processing of the GETE request is performed in a similar manner, but the search will be for the next sequential flight record corresponding to the user specification. Thus, the GETE/GETR routine must maintain a pointer identifying the last block retrieved by the GETE or GETR request executed in the currently active PE.

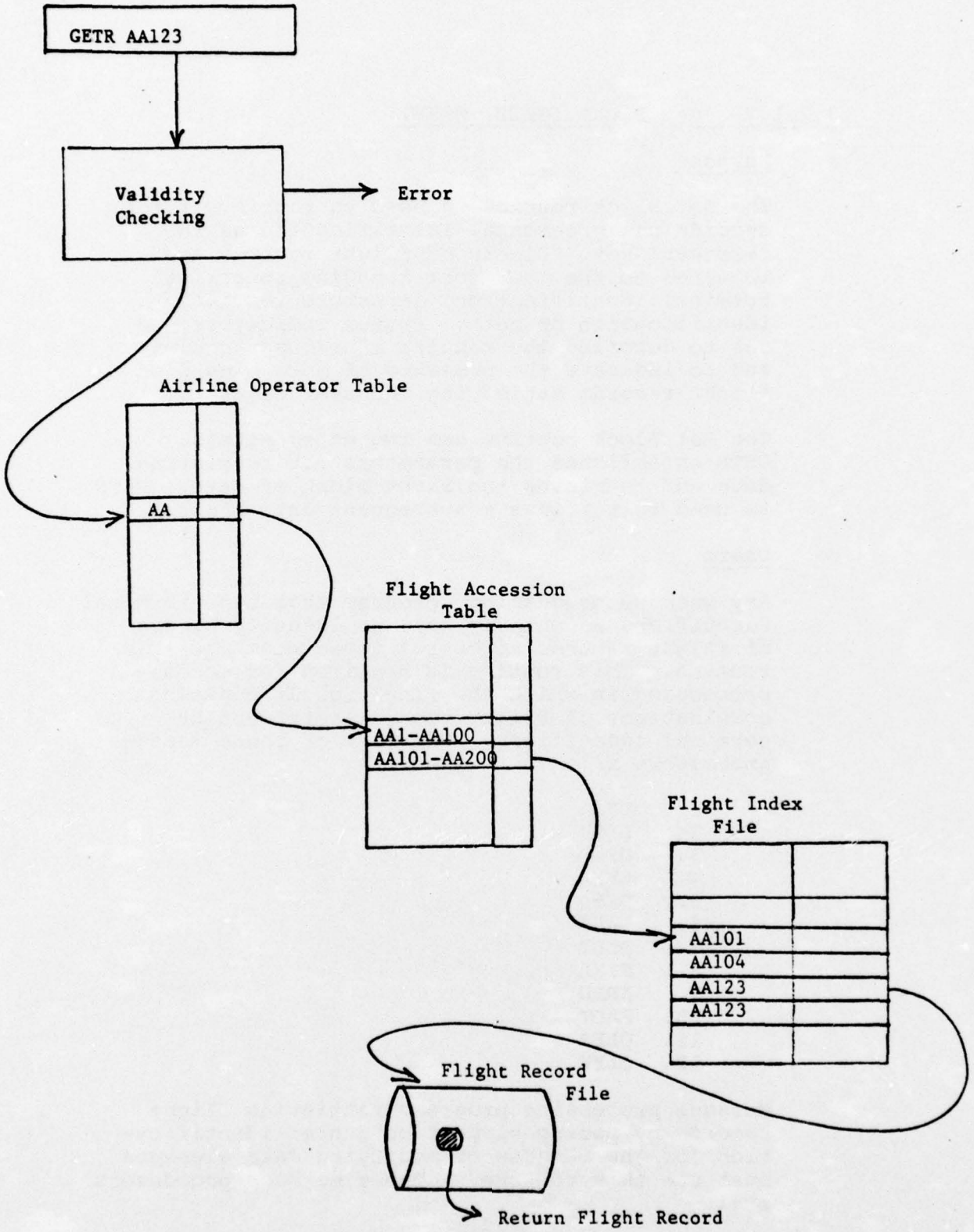


FIGURE 2-4
ILLUSTRATION OF PROCESSING A GETR ROUTINE, WHERE ONLY ACID IS SPECIFIED BY THE USER

2.2.1.2 Get Block (GETB, GETN)

a. Purpose

The Get Block routine is used to retrieve flight records using terminal identification as the retrieval key. Blocks of flight records are returned to the user corresponding to arrival terminal identification, departure terminal identification or both. Status indicators are set to describe the results of request activity and to indicate the presence of more blocks of flight records satisfying the user request.

The Get Block routine has two entry points. GETB establishes the parameters for retrieving data and retrieves the first block of data. GETN is used to retrieve a subsequent data block.

b. Users

Any message processing program that uses terminal identifiers as primary keys to identify blocks of flight records to be retrieved must use this routine. This routine is required for message processing in which the final result contains combinations of flight records retrieved by terminal identifiers. The list of these message processors is:

1. LISA
2. LISD
3. DEMD
4. DEMA
5. DESD
6. DESA
7. DLDY
8. FIXL
9. ARRD
10. FADT
11. QLFZ
12. QLFW

Message processing programs retrieving flight records by pacing airport or center identification for the purpose of modifying data elements must use this routine. Those message processors are:

1. FADP
2. FADF

c. Format of the Calling Sequence

Call GETB WA,

where WA is the location of the start of user request sequence. The data area contains the retrieval parameters, status indicators and the location of user data block.

Call GETN WA,

where WA is the location of user request sequence. The data area contains status indicators and the location of the user data block.

d. Data Segment Definition

1. Data Segment Definition for GETB

1. Arrival/Departure location
2. Flight record qualifier
3. Request status
4. Data block location

2. Data Segment Definition for GETN

1. Request status
2. Data block location

The GETN routine does not require the arrival/departure location or the flight record file, since this information is carried over from the original GNTB request.

These are described below:

1. Arrival/Departure location
The following combinations of arrival/departure information can be specified by the GETB user:

- arrival and departure location
- arrival location only
- departure location only

2. Flight record qualifier

The user may specify one of the desired flight record types:

(i) scheduled flights
(ii) non-scheduled flights
(iii) scheduled and non-scheduled flights

If such is not specified, type (iii) above will be assumed.

3. Request status

The following responses may be provided by DBMS to a GETB or GETN request:

- error in data segment definition or illegal items in the data segment
- no flight record found
- flight record found and placed in data block location
- flight record file from which the block was retrieved
- address of next block satisfying this request

4. Data block location

The memory address of the flight record block.

e. Interface

The tables used in the GETB/GETN routine are:

- Arrival/Departure Table (ADT)

The routine uses disk open, close and read routines. All requests are written to the data recording tape.

f. Processing

Figure 2-5 illustrates the processing of a GETB request specifying only the arrival terminal. Such request will retrieve the first block of flight records of the terminal arrivals. Similarly, the arrival/departure tables will be used to process requests specifying only departure terminals, or request specifying an arrival/departure pair.

The GFTN block always attempts to retrieve the next block satisfying the items in the calling data segment. A pointer is maintained to identify the last block retrieved by the GETB or GETN request executed by the currently active PE.

2.2.1.3 Change Record

a. Purpose

The Change Record routine is used to change data in existing flight records. Flight records are identified by aircraft identifier and the flight leg origin and destination. The flight record data elements that are to be changed are specified by the user as are the elements that do not require a change. The changed data are provided by the user. Status indicators are returned to the user.

b. Users

Any message processing that changes a single flight record must use this routine. The message processing programs using this routine are:

1. FPSD
2. ACTV
3. FP
4. DM
5. INHB

c. Format

Call CHGR WA,

where WA is the location of the start of the user data block containing change parameters and status indicators.

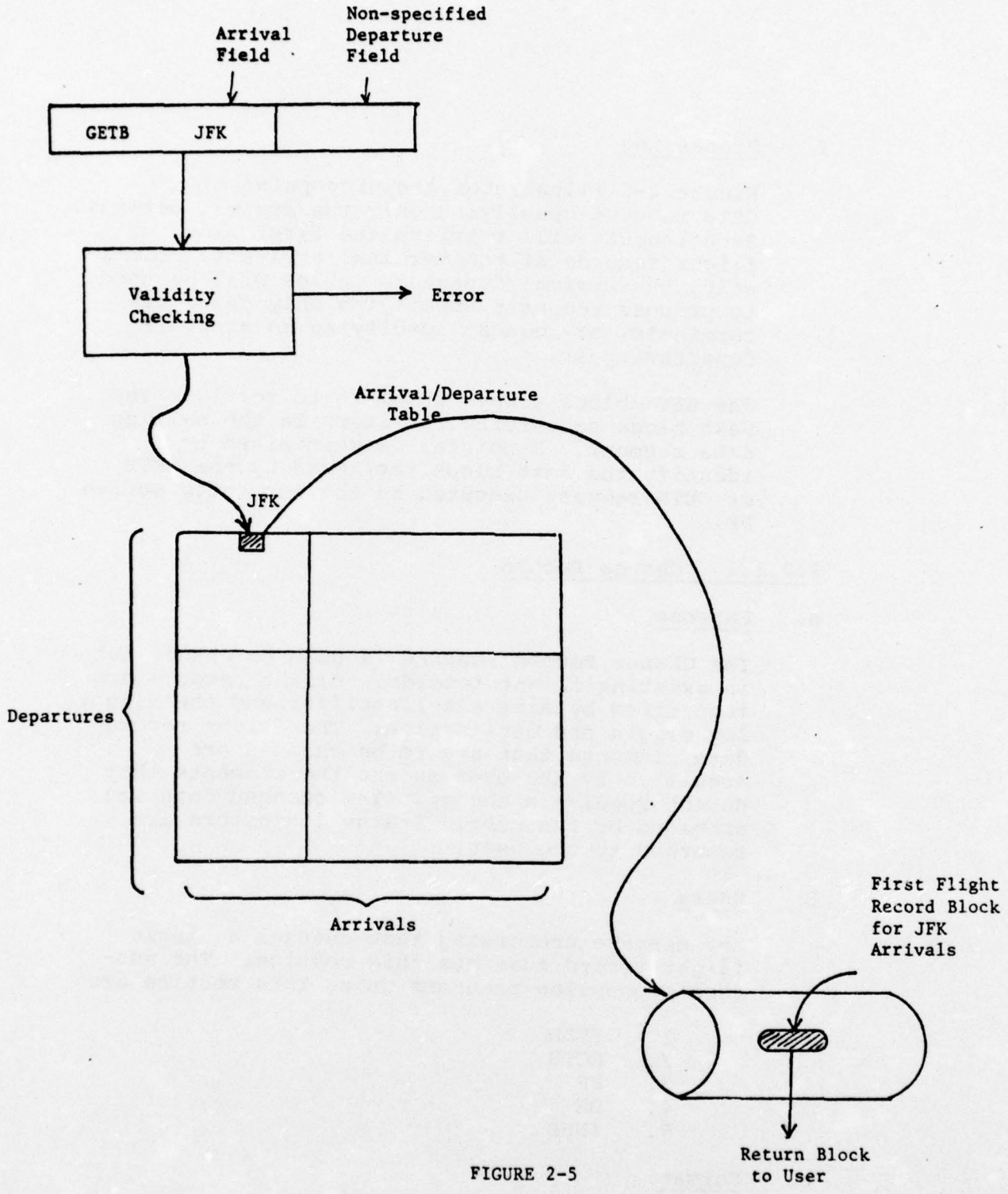


FIGURE 2-5

ILLUSTRATION OF A GETB ROUTINE, WHERE ONLY THE ARRIVAL LOCATION IS SPECIFIED BY THE USER

d. Data Segment Definition

The data segment area contains three groups of data:

Group 1: User Information

This data identifies the user and the file. The following items are provided by the user:

1. caller authorization key (see Section 2.4.1)
2. file to be changed

Group 2: Record Data

The user provides an image (or a "template") of a flight record containing the record identification and the data to be changed. Four data fields must always be provided by the user so that DBMS can properly identify and verify the changed record:

1. aircraft identification
2. relative location ofthe record
3. arrival terminal or center
4. departure terminal or center

The rest of the data fields in the flight record "template" may or may not be provided by the user. Data fields that are not to be changed will be coded as "null". Only the non-null elements will overlay the existing elements in the flight record.

Group 3: DBMS responses

System responses to the CHGR request are included in the data segment. The following responses are possible:

- error in data segment definition or illegal items in the data segment
- flight record was found and was changed

- caller is not authorized to perform this CHGR operation
- flight record identification does not match with the flight record retrieved

e. Interfaces

System routines uses are disk open, close, write, lock and unlock. All requests are written to tape.

f. Processing

The processing associated with the CHGR routine is outlined as follows:

1. check the validity of the CHGR data segment and the items in the data segment.

2. retrieve the record based on the relative location provided in the data segment.

3. check if the ACID, arrival and departure terminals match with the user identification of the flight record. If not - the CHGR request is rejected.

4. change the appropriate elements, and replace the record on disk.

5. update pointers and flags.

6. return status information to the user.

2.2.1.4 Change Block

a. Purpose

The Change Block routine is used to change data elements in a block of flight records. The block of flight records is selected by an arrival terminal identifier; individual flight records are located by aircraft identifier, departure terminal and relative location. The user prepares an image (or a "template") of a block,

which identifies the logical records to be changed the data elements to be changed and accompanied by the desired changes. Status indicators are returned to the user.

b. Users

Any message processing that requires changes to data elements in multiple flight records should use this routine. The flight records are identified with an arrival terminal. The message processing programs using this routine are:

1. FADP
2. FADF

c. Format

Call CHGB WA,

where WA is the location of the start of the user data containing location parameters, status indicators, record count and the location of change records.

d. Data Segment Definition

The data segment area contains three groups of data.

Group 1: User Information

This data identifies the user and the file. The following items are provided by the user:

1. caller authorization key
2. file to be changed

Group 2: Block Data

The user provides an impage (or a "template") of a flight record containing the record identification and the data to be changed. For each record to be changed in the block the following data elements must be present:

1. aircraft identification
2. relative location of the record
3. arrival terminal or center
4. departure terminal on center
5. the "change record" flag must be set

Data fields that are not to be changed will be coded as "null" fields. Only the non-null elements in the changed records will overlay the existing elements in the file.

Group 3: DBMS resources

The system may return the following status information:

- error in data segment definition or illegal items in the data segment
- change block not located
- caller not authorized to perform this CHGB operation
- flight record identification does not match with the flight record retrieved
- next block for this arrival terminal (or no more blocks)
- request completed
- number of records changed

e. Interfaces

System routines used are: disk open, close, write, lock and unlock. All requests are written to the data recording tape.

f. Processing

The processing associated with the CHGB routine is outlined as follows:

1. check the validity of the CHGB data segment and the items in the data segment

2. retrieve the block based on the block addressing information provided by the user in the data segment

3. for each record to be changed, as indicated by the "change record" flag, check if the ACID, arrival and departure terminals match with the user identification of the flight record. If not - reject this request

4. update the block as requested and replace the block on disk

5. update pointers and flags

6. return status information to the user

2.2.1.5 Insert Record

a. Purpose

The Insert Record routine is used to create a new flight record for scheduled and non-scheduled flights. The user supplies all information necessary to identify the flight record and to generate a minimum subset of data elements. The data elements that have to be defined by the user are specified in Section d. below. The flight record is added to the data base and is included in the index information. Status indicators are returned to the users.

b. Users

Any message processing that creates new flight records must use this routine. The messages using this routine are:

1. FPSD
2. FP

c. Format

Call INST WA,

where WA is the location of the start of user request sequence containing flight record parameters and status indicators.

d. Data Segment Definition

The data segment area contains three groups of data:

Group 1: User information

This data includes:

1. caller authorization key
2. file to be changed

Group 2: Record data

The user provides an image (or a "template") of a flight record containing the new data items. Some of the data field are optional and may be coded as "null"; but the following data must be provided by the user:

1. Aircraft identifier
2. Arrival terminal
3. Departure terminal
4. Scheduled departure time
5. Estimated time enroute
6. Departure date mask
7. Arrival date mask

Group 3: DBMS responses

The system may return the following status information:

- error in data segment definition or illegal items in the data segment

- data provided not sufficient to accomplish the INST request
- caller not authorized to perform this function
- request completed - relative record address is returned

e. Interfaces

System routines used are: disk open, close, read, write, lock and unlock. All requests are written to the data recording tape.

f. Processing

The processing associated with the INST routine is outlined as follows:

1. Check the validity of the INST data segment and the items in the data segment.

2. Create a flight record constructed from the user defined data. If essential elements are missing or inconsistent - return an error status code.

3. Look up the Arrival/Departure Table corresponding to this flight record.

4. Retrieve the corresponding Arrival/ Departure block from disk.

5. Insert the record in the Arrival/ Departure block and update the appropriate header information. If there is an overflow block - retrieve it and update appropriately. If a new overflow block is required - generate it. Write the updated block (or the user block) back on disk.

6. Update the system tables (e.g., Flight Index File, Flight Accession Table, etc.), pointers and flags.

7. Return status code to the user.

2.2.1.6 Remove Record

a. Purpose

The Remove Record routine is used to delete references to flight records. Neither the flight records nor the index information is deleted from the data base, but flags are set to indicate that the flight record is no longer active. The user must supply information to uniquely identify the flight record that is to be deleted. Status indicators are returned to the user.

b. Users

Any message processing that deletes references to existing flight records should use this routine. The messages using this routine are:

1. CXSD
2. RS

c. Format

Call REMR WA,

where WA is the location of the start of user request sequence containing the flight record reference parameters and status indicators.

d. Data Segment Definition

The data segment area contains three groups of data:

Group 1: User information

This data includes:

1. caller authorization key
2. file to be affected

Group 2: Record data

The user provides an image of a flight record containing the following information identifying the record to be changed:

1. aircraft identification
2. relative location of the record
3. arrival terminal or center
4. departure terminal or center

Group 3: DBMS responses

System responses to the REMP request are includes in the data segment. The following responses are possible:

- error in data segment definition or illegal items in the data segment
- flight record not found
- caller is not authorized to perform this REMR operation
- flight record identification does not match with the flight record retrieved
- flight record was found and was deleted

e. Interfaces

System routines used are: disk open, close, read, write, lock and unlock. All requests are written to the data recording tape.

g. Processing

The processing associated with the REMR routine is outlined as follows:

1. check the validity of the REMR data segment and the items in the data segment.

2. retrieve the record based on the relative location provided by the user in the data segment.

3. check if the ACID, arrival and departure terminals match with the user identification of the flight record. If not - the REMR request is rejected.

4. flag the record as "deleted", update the appropriate pointers and flags and write the deleted record back on disk.

5. Update system tables, pointers and flags.

6. return status information to the user.

2.2.2 User Access to General Tables

The GET TABLE (TABT) and SET TABLE (SETT) routines provide the applications programs with the capability to read and write those general tables which directly interface with the user.

2.2.2.1 Get Table

a. Purpose

The Get Table routine is used to retrieve table values using the table identifier as the retrieval key. Subsequent identifiers are table dependent. The table values corresponding to the identifiers are returned to the user, accompanied by status information.

b. Users

Any message processing requiring access to general tables should use this routine. The tables accessible through the TABT request are:

1. Capacity table (CAT)
2. General aviation table (GAT)
3. Aircraft class table (ACT)
4. Aircraft type table (ATT)
5. Airline operator table (AOT)
6. Flight accession table (FAT)
7. Zone table (ZOT)
8. Arrival Fix table (FXT)

9. Airport/fix table (AFT)
10. Airport table (APT)
11. Center table (CET)
12. Conversion dictionary table (CDT)
13. Continue table (COT)
14. Operational category table
15. Output format table
16. Output device table
17. Parameter table
18. Non-OAG name table

Some message processing requires access to the tables by virtue of the message itself. These messages are:

1. CAPL
2. GAEL

Other messages use the tables in a supplemental mode, such as checking the validity of message elements.

c. Format

Call TABT WA,

where WA contains the table identifier and retrieval parameters.

d. Data Segment Definition

1. Table identification
2. First qualifier (table type dependent)
3. Second qualifier (table type dependent)
4. Status indicators
   - error in data segment
   - table entry not found
   - table entry was found and was placed in the user's work area
5. location of user work area

The first and second qualifiers for each table accessed are defined below:

Capacity Table

Arrival pacing airport
Not used

General Aviation Table

Pacing airport or center
Not used

Aircraft type table

Aircraft type
Not used

Aircraft class table

Aircraft class
not used

Airline Operator Table

Operator
Not used

Flight Accession Table

Not used
Not used

Zone Table

Packing airport
Not used

Arrival Fix Table

Fix identifier
Not used

Airport/fix Table

Pacing airport
Not used

Airport Table

Airport
Not used

Center Table

Center
Not used

Conversion Dictionary Table

Conversion value
Not used

Continue Table

First simulation qualifier
Second simulation qualifier

Operational Category Table

Not used
Not used

Output Format Table

Not used
Not used

Output Device Table

Not used
Not used

Parameter Table

Parameter ID
Not used

Non-OAG Name Table

Not used
Not used

e. Interfaces

None

f. Processing

The processing associated with the TABT routine is outlined as follows:

1. Check the validity of the TABT data segment and the items in the data segment.

2. Retrieve the requested data based on the identification provided by the user.

3. Return status information to the user.

2.2.2.2 Set Table

a. Purpose

The Set Table routine is used to set values into user controlled areas in the specified tables. The specific table is identified by table code; subsequent values are used to define the changed data.

b. Users

Any message processing requiring changes to the general tables must use this routine. Tables that can be changed by the user are:

- Capacity Table (CAT)
- General Aviation Table (GAT)

The remaining tables cannot be altered by the user, and no message exists to change the values in the tables.

The messages that use this routine are:

1. CAPS
2. GAES

c. Format

Call SETT WA,

where WA contains the table identifier, retrieval parameters and table value.

d. Data Segment Definition

1. Table identifier
2. Caller authorization key
3. First qualifier - arrival terminal
4. Second qualifier - time
5. Status indicators:
   - error in data segment
   - table entry not found
   - user not authorized to perform this function
   - rejected due to unreasonable values
   - request was accomplished
6. New table value

e. Interfaces

None

f. Processing

The processing associated with the SETT routine is outlined as follows:

1. check the validity of the SETT data segment and the items in the data segment
2. check if the requested changes are reasonable. If not - reject
3. perform the requested changes
4. return status information

2.2.3 DBMS Access to the Data Base

The Logical File Handlers (LFHs) described in Sections 2.2.1 and 2.2.2 provide the interface between the application programs and the Data Base components accessed by the application programs. Additional LFHs are required to facilitate the internal functions associated with DBMS processing of the Data Base. These LFHs are described in this section.

2.2.3.1 Get Index Record

a. Purpose

The Get Index Record routine is used to obtain index records using aircraft identification as the retrieval key. The block of flight index records containing the first reference to the aircraft identification is returned to the user. Status indicators are used to describe the results of request activity.

b. Users

The Get Index Record routine is used by the Logical File Handlers and is not available to applications programs. The routine should be used in retrieving flight records by aircraft identification. The Logical File Handlers using this routine are:

1. Get record
2. Insert record

c. Format

Call GIND WA,

where WA is the location of the start of the data block containing aircraft identification, status indicators and the location of the data block.

d. Data Segment Definition

1. Aircraft Identification

2. Data Location

The memory address where the flight index records will be placed.

3. Request status

Provides system responses

e. Interfaces

The GIND uses the Airline Operator Table (AOT), the Flight Accession Table (FAT) and the Flight Index File (FIF). System routines used are disk open, close, and read.

f. Processing

The processing associated with the GIND routine is outlined as follows:

1. perform general validation checks of this request

2. look up the AOT based on the operator code in the requested data segment and get the pointer to FAT.

3. look up FAT to find the appropriate accession code as in the requested data segment and get the pointer to FIF.

4. retrieve the FIF record and place the data in the user defined work area.

5. provide the appropriate responses in the status data field.

2.3 Executive Interface Facilities

The DBMS requires Executive services in order to access the data stored on disk. Communications between DBMS and the Executive is accomplished through Supervisor Calls (SVCs). A data block is established to pass control information to the Executive. Another block is used to pass data to and from the data base. DBMS must provide for sufficient data buffer storage.

2.3.1 Executive Services

The LFHs invoke various requests to be serviced by the Executive. The Executive functions used by the LFHs are:

- Allocate free storage (Allocate)
- Release free storage (Release)
- Open operations on the disk (Open)
- Close operations on the disk (Close)
- Lock out other disk operations (Lock)*
- Release lock out of other disk operations (Unlock)
- Read from the disk (Read)
- Write to the disk (Write)

In addition to these services, Executive procedures record transactions, maintain data base back-up, provide audit, historical and statistical recordings and gather data for off-line data reduction.

2.3.2 Data Base Retrieval

A retrieval request generates the following functional steps by DBMS:

- Allocate
- Open
- Read
- Close
- Release

---

* The Lock operation prohibits any access to a specific portion of the disk (e.g., track).

2.3.3 Data Base Update

An update request generates the following functional steps by DBMS:

- Allocate
- Open
- Read and Lock
- Write and Unlock
- Close
- Release

Normally, an update operation will be accomplished by an initial READ/LOCK request, followed by some processing to build an updated block of data, and finally issuing a WRITE/UNLOCK request.

2.3.4 Data Base Lock/Unlock

The update request causes the data base system to prevent access to a portion of the data base until the update is complete. When the update is finished, the restriction is lifted. The Executive functions are Lock and Unlock.

2.4 Control and Utility Functions

In addition to the LFHs which provide the basic mechanism for user interface with the data base, the DBMS must perform various functions which normally are not directly accessible or visible to the user.

These functions perform data base security checks, initiate start-up and start-over processing, and support testing and evaluation of the data base subsystem.

2.4.1 Security

The DBMS must guarantee against inadvertent and unauthorized changes to the data base. The LFHs which are available to the applications programs for the purpose of making changes to the data base contents are:

- Change Record (CHGR)
- Change Block (CHGB)
- Insert Record (INSR)
- Remove Record (REMR)
- Set Table (SETT)

In order to protect the data base, the caller of the above LFHs shall be required to provide, within his calling sequence, an "Authorization Key" which uniquely identifies whether or not the calling program is authorized to perform the requested changes in the data base. The status code field will return a reject code if the caller is unauthorized. The Authorization Keys will be distributed and controlled by the Chief Programmer.

2.4.2 Recovery Recording and Start-Over

At equal time intervals, the complete data base shall be read and saved on magnetic tape. The primary use of the recovery recording on tape will be to support start-over processing as described below. Other uses might be to provide the input for data reduction and analysis performed off-line, and to assist in testing and debugging.

Start-over processing is initiated upon detection of errors or destroyed data in the data base which prevent the orderly continuation of system operations. Such conditions are determined by the Executive, or by the Data Base Management Subsystem. In addition, the operator may externally invoke start-over processing.

In this context, data base start-over should not be confused with system start-over; the latter being a complete system recovery from an adverse condition. The data base start-over involves only the regeneration of the data base, without impacting the rest of the system other than the time needed to accomplish the start-over. Data base start-over may or may not be necessary as part of a total system start-over.

Two modes of data base start-over shall be possible: basic start-over and advanced start-over. These are described below.

In the basic start-over mode the current data base is replaced by the last recorded data base on tape. All the data base updates which may have occurred in the interim will be ignored.

The advanced start-over mode will attempt to regenerate the destroyed data base by updating the last recorded data base according to the update functions which have been accomplished and which are not represented on the last recorded data base.

2.4.3 Data Base Logs

In this system, the bulk of on-line recording is accomplished by the Executive. Significant events and various data are recorded on tape cyclically or on demand, and subsequently analyzed and reduced off-line. While it is the Executive who performs the actual recording, the DBMS communicates to the EXEC the events which are to be recorded. These events are listed below:

- Start data base access processing

  The recorded data will include all information associated with the calling data segment, including time and caller identification.

- Complete data base access processing

  The recorded data will include the type of access accomplished, the time, the completion status code, the record count and the caller identification.

- Changes to the data base

  The recorded data will include information on changes that were implemented affecting the data base contents.

2.4.4 Data Base Test and Evaluation Function (DBTEF)

The DBTEF provides the basic tools necessary in order to display and modify data base information. These actions shall be initiated through the operator keyboard device. The output will also be displayed at the operator console.

2.4.4.1 Display

The Display routine is used to dump sections of the data base. The format is:

DSPLY (qualifiers...)

The qualifiers are:

1. File identifier
2. Start relative address
3. Word count

2.4.4.2 Change

The change routine is used to change data base information. The format is:

CHNG (qualifiers...)

The qualifiers are:

1. File identifier
2. Start relative address
3. New data

2.4.5 Start-Up

The Start-up function of DBMS is responsible for the initial processing associated with reading in the off-line prepared data, checking for the completeness of the data, building the indexing tables, setting up of the required linkages and placing the data in its final disk and/or main memory storage areas.

2.4.6 Pointer Maintenance

Utility routines must be provided to enable the DBMS to properly and efficiently update the pointers used for cross-referencing between the data base tables. Each of these Pointer Maintenance Routines will be individually responsible for updating a specific file (e.g., Flight Index File, Arrival/ Departure Table, etc.).

2.4.7 Statistics Collection

The following performance measure statistics (e.g., mean, max, running total) will be maintained by DBMS:

- statistics on queue length for disk access
- statistics on queue wait for disk access
- statistics on disk reads
- statistics on disk writes
- statistics on disk lock and unlock

These statistics will be periodically recorded on the data recording tape. In addition, they will be displayed at the System Monitoring Position (SMP) in response to a supervisory or a system programmer request.

2.4.8 On-line Reporting

The following data-base-related events will cause an appropriate display or a message printout at the System Monitoring Position (SMP):

- unauthorized request for data base update
- data base buffer area exceeded
- response time for a data base access exceeded a predefined (SP) time

APPENDIX A

DATA BASE SUBSYSTEM - OFF-LINE/ON-LINE COORDINATION

The purpose of this Section is to specify the relationship between the off-line and the on-line systems relative to the Data Base Subsystem. In general, the on-line system is concerned with updating and maintaining the data base in an operational environment and in real time. Additionally, the on-line system provides data recorded during system operation for subsequent reduction by the off-line system. The off-line system is also responsible for the initial building of the data base, and for providing an analysis of new and "used" data bases.

The following list describes the data base-related functions which shall be performed by the off-line system:

1. Initial Data Base Build

   The off-line system analyzes the OAG and other source data and builds the data base files and tables as specified in Section 1. All the necessary linkages, cross references, pointer fields shall be supplied by the off-line system in a form ready to be used by the on-line system. Data fields which are initially assigned by the on-line system (e.g., data fields in NSFRF) will be designated as "null" data by the off-line system.

2. Off-line Data Base Update

   The off-line system shall have the capability to update an existing data base according to user-supplied source data.

   Such an update may provide the following functions:

   1. insert or remove flight records
   2. perform garbage collection on a "used" data base (e.g., remove records flagged as deleted)
   3. make changes to the General Tables

3. Data Base Printout

The off-line system shall provide, at the user option, a readable printout of the entire data base or portions thereof. The data base thus printed may be a new one just built or it may be a "used" data base recorded by the on-line system.

4. Data Base Analysis

The off-line system shall have the capability to perform data reduction and analysis of a new or "used" data base. The following output shall be provided:

- number of flight records for each arrival terminal, each departure terminal and each arrival/departure pair.
- total number of flight records
- statistics on number of flight records in physical records on disk

5. Data Base Compare

The off-line system shall be able to compare two "used" data bases and provide an analysis of their differences.

6. Data Base Access Analysis

The off-line system shall provide, at the user option, an analysis of the on-line recorded logs of events associated with data base access. The following statistics shall be available:

- statistics on data base accesses by type and by originator
- statistics on utilization of the various data base files and tables
- statistics on data base queues, buffers, and response times

APPENDIX B

USE OF LOGICAL FILE HANDLERS (LFHs) FOR MESSAGE PROCESSING

This section summarizes user reference information describing how the application programs make use of the Data Base processing features provided through the LFHs. The following operations are outlined:

1. list flight plans
2. create flight plan
3. delete flight plan
4. change flight plan
5. retrieve pacing airport or center data
6. change pacing airport data

1. List Flight Plans

Listing flight plans use only the Get Record LFH along with its second entry point Get Next Record. Flight records are retrieved by one of the following:

a. Aircraft identification only
b. Aircraft identification;
   arrival pacing airport or center
c. Aircraft identification;
   departure pacing airport or center
d. Aircraft identification;
   arrival pacing airport or center;
   departure pacing airport or center

A qualifier aids in the retrieval process by specifying:

a. Scheduled operations only
b. Non-scheduled operations only
c. Scheduled and non-scheduled operations

Processing steps are:

a. Call Get Record with request list and to retrieve first flight record;
b. Call get Next Record for subsequent records;
c. Assemble list of flight records according to request.

2. Create Flight Plan

Creating flight plans use Get Record, Get Next Record, Insert Record and Change Record entry points in the Logical File Handlers. Flight records are retrieved by one of the following:

a. Aircraft identification; departure pacing airport or center
b. Aircraft identification; arrival pacing airport or center; departure pacing airport or center

A qualifier aids the retrieval process by specifying:

a. Scheduled operations only (by FPSD)
b. Non-scheduled operations only (by FP)

Mode of entry into flight plan processing enables the setting of the qualifier.

Processing steps are:

a. Call Get Record with request list and to retrieve first flight record;
b. Call Get Next Record for subsequent records;
c. If flight record found, call Change Record with appropriate changes;
d. Otherwise, call Insert Record with new flight record values.

3. Delete Flight Plan

Deleting flight plans use Get Record, Get Next Record and Remove Record entry points in the Logical File Handlers. Flight plans are retrieved by one of the following:

a. Aircraft identification; departure pacing airport or center
b. Aircraft identification; arrival pacing airport or center; departure pacing airport or center.

A qualifier aids the retrieval process by specifying:

a. Scheduled operations only (by CXSD)
b. Non-scheduled operations only (by RS)

Mode of entry into flight plan processing enables the specifying of the qualifier.

Processing steps are:

a. Call Get Record with request list and to retrieve first flight record;
b. Call Get Next Record for subsequent records;
c. When flight record found, call Remove Record to cause the record to be de-activated.

4. Change Flight Plan

Changing flight plans use Get Record, Get Next Record and Change Record entry ponts in the Logical File Handlers. Flight plans are retrieved by one of the following:

a. Aircraft identification,
b. Aircraft identification departure pacing airport or center.

A qualifier aids the retrieval process by specifying:

a. Scheduled operations only
b. Non-scheduled operations only

Inspection of the operator identification of the aircraft identification enables the specifying of the qualifier.

Processing steps are:

a. Call Get Record with record request list and to retrieve first flight record,
b. Call Get Next Record for subsequent records
c. When flight record found, call Change Record with change request sequence.

5. Retrieve Pacing Airport or Center Data

Retrieving pacing airport or center data use Get Block and Get Next Block entry points in the Logical File Handler. Blocks of data are retrieved by one of the following:

a. Arrival pacing airport or center
b. Departure pacing airport or center

c. Arrival pacing airport or center
departure pacing airport or center

Qualifier aid the retrieval process by specifying:

a. Scheduled operations only,
b. Non-scheduled operations only,
c. Scheduled and non-scheduled operations.

Processing steps are:

a. Call Get Block with request list and to retrieve first block of data,
b. Call Get Next Blocks for subsequent blocks of data.

6. Change Pacing Airport Data

Changing data for pacing airports use the Change Block entry point in the Logical File Handler. Only one format is acceptable by this handler:

. Pacing airport

No qualifiers are allowed.

The list of a block of flight records to be changed are submitted by the request. All modifications are performed by the handler.